Elastix Web Services (WSDL)
Especificaciones Funcionales

Elaborado por: Departamento de Desarrollo de Elastix
Versión: Elastix 2.0.4-Beta 2

VERSIONAMIENTO

Versión	Versión de Elastix	Fecha	Editado por	Aprobado Por	Detalles
1	2.0.4-beta2	20-Abr-2011	Ing. Alberto Santos	Ing. Bruno Macias	Documentación Inicial

Elastix Web Services en Elastix 2.0.4-*

Generalidades

En la actualidad la obtención y manipulación de datos es el centro funcional de cualquier tipo de empresa. Es por esto que hemos decidido crear Web Services para Elastix, de esta manera podremos obtener o manipular los datos de nuestro servidor Elastix desde diversos escenarios.

Objetivo

Crear Web Services para Elastix que nos permitan obtener o modificar información importante de ciertos módulos del servidor Elastix. Dicha información debe pertenecer al usuario autenticado.

Herramientas de Desarrollo

Todas las funcionalidades de los Elastix Web Services han sido desarrolladas en PHP.

Funcionalidad

A continuación se detallan los módulos a los que se crearon Web Services:

Módulo Calendar

- Creación de un Web Service para el módulo “calendar” al cual se le incluirán los siguientes métodos:
    - **listCalendarEvents:** Lista los eventos del módulo calendar pertenecientes al usuario autenticado y en el rango de fecha establecido.
      Parámetros de entrada:
        - startdate: (date) fecha inicial del evento.
        - enddate: (date) fecha final del evento.

Parámetros de salida:

- id: (positiveInteger) id del evento en la base de datos.
- startdate: (date) fecha de inicio del evento.
- enddate: (date) fecha en que finaliza el evento.
- starttime: (datetime) hora de inicio del evento.
- endtime: (datetime) hora en que finaliza el evento.
- subject: (string) asunto del evento.
- description: (string) descripción del evento.
- asterisk_call: (bool) true si debe ser generada una llamada de recordatorio.
- recording: (string, opcional) nombre del archivo de grabación usado para la llamada.
- call_to: (string, opcional) extensión a la cual se llamará para el recordatorio.
- reminder_timer: (string, opcional) número de minutos antes de que la llamada de recordatorio se realice.
- emails_notification: (array(string)) emails que serán notificados con un mensaje cuando se crea el evento.

◦ **addCalendarEvent:** Agrega un nuevo evento al calendar del usuario autenticado.

Parámetros de entrada:

- startdate: (date) fecha de inicio del evento.
- enddate: (date) fecha en que finaliza el evento.
- subject: (string) asunto del evento.
- description: (string) descripción del evento.
- asterisk_call: (bool) true si debe ser generada una llamada de recordatorio.
- recording: (string, opcional) nombre del archivo de grabación usado para la llamada.
- call_to: (string, opcional) extensión a la cual se llamará para el recordatorio.
- reminder_timer: (string, opcional) número de minutos antes de que la llamada de recordatorio se realice.
- emails_notification: (array(string)) emails que serán notificados con un mensaje cuando se crea el evento.
- color: (string, opcional) color que tendrá el evento.

Parámetros de salida:

- return: (bool) true si el evento fue creado, caso contrario false.

◦ **delCalendarEvent:** Elimina un evento perteneciente al usuario autenticado.

Parámetros de entrada:

- id: (positiveInteger) id del evento que será eliminado.

Parámetros de salida:

- return: (bool) true si el evento fue eliminado, caso contrario false.

- Creación de un Web Service para el módulo “address_book” al cual se le incluirán los siguientes métodos:

Módulo AddressBook

  - **listAddressBook:** Lista los contactos que tiene ingresado el usuario autenticado.
    Parámetros de entrada:
    - addressBookType: (string) tipo de libreta, puede ser “internal” o “external”.
    - offset: (positiveInteger, opcional) inicio de registros, o 0 si es omitido.
    - limit: (positiveInteger, opcional) límite de registros o todos si es omitido.

    Parámetros de salida:
    - totalCount: (positiveInteger) total de registros.
    - extension: (array) arreglo con los siguientes datos:
      - id: (positiveInteger) id del contacto en la base de datos.
      - phone: (string) teléfono del contacto.
      - name: (string) nombre completo del contacto.
      - first_name: (string, opcional) primer nombre del contacto.
      - last_name: (string, opcional) apellido del contacto.
      - email: (string, opcional) email del contacto.

  - **addAddressBookContact:** Agrega un nuevo contacto para la lista “external” del usuario autenticado.
    Parámetros de entrada:
    - phone: (string) teléfono del contacto.
    - first_name: (string, opcional) nombre del contacto.
    - last_name: (string, opcional) apellido del contacto.
    - email: (string, opcional) email del contacto.

    Parámetros de salida:
    - return: (bool) true si el contacto fue agregado, caso contrario false.

  - **delAddressBookContact:** Elimina un contacto de la lista “external” del usuario autenticado, además borra cualquier archivo relacionado a ese contacto.
    Parámetros de entrada:
    - id: (positiveInteger) id del contacto a ser eliminado.

    Parámetros de salida:
    - return: (bool) true si el contacto fue eliminado, caso contrario false.

- Creación de un Web Service para el módulo “cdrreport” al cual se le incluirán los siguientes métodos:

  - **listCDR:** Lista las llamadas realizadas y recibidas que estén relacionadas al usuario autenticado.

Parámetros de entrada:

- startdate: (date, opcional) fecha de inicio para los registros, o todos si es omitido.
- enddate: (date, opcional) fecha máxima para los registros, o todos si es omitido.
- offset: (positiveInteger, opcional) inicio de registros, o 0 si es omitido.
- limit: (positiveInteger, opcional) límite de registros o todos si es omitido.

Parámetros de salida:

- totalcdrcount: (positiveInteger) número total de registros.
- cdrs: (array) arreglo con los siguientes datos:
  - calldate: (datetime) fecha y hora de la llamada.
  - src: (string) extensión origen en la llamada.
  - dst: (string) extensión destino en la llamada.
  - channel: (string) canal donde se originó la llamada.
  - dstchannel: (string) canal destino de la llamada.
  - disposition: (string) descripción del estado de la llamada.
  - uniqueid: (string) id de la llamada.
  - duration: (positiveInteger) duración en segundos de la llamada.
  - billsec: (positiveInteger) duración en segundos de la llamada a cobrar.
  - accountcode: (string) código de la cuenta.

Módulo Fax

- Creación de un Web Service para el módulo “faxviewer” al cual se le incluirán los siguientes métodos:
  - **listFaxDocs:** Lista los faxes enviados y recibidos por el servidor Elastix.

    Parámetros de entrada:
    - date: (date, opcional) fecha en que los faxes fueron enviados o recibidos, o todos si es omitido.
    - direction: (string, opcional) 'in' para faxes recibidos y 'out' para faxes enviados, o todos si es omitido.
    - offset: (positiveInteger, opcional) inicio de registros, o 0 si es omitido.
    - limit: (positiveInteger, opcional) límite de registros o todos si es omitido.

    Parámetros de salida:
    - totalfaxcount: (positiveInteger) número total de registros.
    - faxes: (array) arreglo con los siguientes datos:
      - id: (positiveInteger) id del fax en la base de datos.
      - Modemdev: (string) dispositivo TTY del fax enviado o recibido.
      - errormsg: (string, opcional) mensaje de error en caso de existir.
      - company_name: (string) descripción de la compañía.
      - company_fax: (string) descripción del número de la compañía.
      - date: (datetime) fecha y hora del fax.
      - type: (string) 'in' para faxes recibidos y 'out' para faxes enviados.
      - destiny_name: (string) nombre del dispositivo fax destino.
      - destiny_fax: (string) número del fax destino.

- **delFaxDoc:** Elimina un fax, y el documento PDF asociado a él en caso de que exista.
  Parámetros de entrada:
  - id: (positiveInteger) id del fax que será eliminado.

  Parámetros de salida:
  - return: (bool) true si el fax fue eliminado, caso contrario false.

Módulo My_Extension

- Creación de un Web Service para el módulo “my_extension” al cual se le incluirán los siguientes métodos:
  - **setCallWaiting:** Habilita la opción de call waiting para la extensión del usuario autenticado.
    Parámetros de entrada:
    - callWaiting: (bool) true para activar el call waiting, caso contrario false.

    Parámetros de salida:
    - return: (bool) true si se activó el call waiting, caso contrario false.

  - **setCallMonitor:** Habilita las opciones de call monitor para la extensión del usuario autenticado.
    Parámetros de entrada:
    - recordIncoming: (string) ésta opción puede ser 'Always', 'Never' o 'Adhoc'. Se habilita para llamadas entrantes.
    - recordOutgoing: (string) ésta opción puede ser 'Always', 'Never' o 'Adhoc'. Se habilita para llamadas salientes.

    Parámetros de salida:
    - return: (bool) true si las opciones de call monitor fueron habilitadas, caso contrario false.

  - **setDoNotDisturb:** Habilita la opción de “do not disturb” para la extensión del usuario autenticado.
    Parámetros de entrada:
    - doNotDisturb: (bool) true para habilitar “do not disturb”, caso contrario false.

    Parámetros de salida:
    - return: (bool) true si la opción de “do not disturb” fue habilitada, caso contrario false.

  - **setCallForward:** Habilita las opciones de “call forward” para la extensión del usuario autenticado.
    Parámetros de entrada:
    - callForward: (bool) true para habilitar la opción de “call forward”, caso contrario false.
    - phoneNumberCallForward: (positiveInteger) número para la opción “call forward”.
    - callForwardUnavailable: (bool) true para habilitar la opción de “call forward on unavailable”, caso contrario false.

- phoneNumberCallForwardUnavailable: (positiveInteger) número para la opción “call forward on unavailable”.
- callForwardBusy: (bool) true para habilitar la opción de “call forward on busy”, caso contrario false.
- PhoneNumberCallForwardBusy: (positiveInteger) número para la opción “call forward on busy”.

Módulo Voicemail

- Creación de un Web Service para el módulo “voicemail” al cual se le incluirán los siguientes métodos:
  - **listVoicemail:** Lista los voicemails que tiene la extensión del usuario autenticado.

    Parámetros de entrada:
    - startdate: (date) fecha inicio de los registros.
    - enddate: (date) fecha final de los registros.

    Parámetros de salida:
    - totalVoicemail: (positiveInteger) número total de registros.
    - voicemail: (array) arreglo con los siguientes datos:
      - date: (date) fecha del voicemail.
      - time: (date) hora del voicemail.
      - callerid: (string) extensión a la que le pertenece el voicemail.
      - extension: (string) extensión que generó el voicemail.
      - duration: (positiveInteger) duración en segundos del voicemail.

  - **delVoicemail:** Elimina un voicemail asociado a la extensión del usuario autenticado.

    Parámetros de entrada:
    - file: (string) nombre del archivo voicemail.

    Parámetros de salida:
    - return: (bool) true si el voicemail fue eliminado, caso contrario false.

  - **setConfiguration:** Realiza la configuración del voicemail de la extensión asociada al usuario autenticado.

    Parámetros de entrada:
    - enable: (bool) true para habilitar la configuración del voicemail, caso contrario false.
    - email: (string) email para el voicemail.
    - pagerEmail: (string, opcional) email secundario.
    - password: (string) clave para el voicemail.
    - confirmPassword: (string) confirmación de clave, debe ser igual a password.
    - emailAttachment: (bool) true para encender la opción de email attachment, caso contrario false.
    - playCID: (bool) true para encender la opción de play CID, caso contrario false.
    - playEnvelope: (bool) true para encender la opción de play envelope, caso

contrario false.
        - deleteVmail: (bool) true para encender la opción de delete Vmail, caso contrario false.

      Parámetros de salida:
        - return: (bool) true si la configuración fue realizada, caso contrario false.

    - **downloadVoicemail:** Permite obtener el tamaño, el tipo de contenido y el contenido en sí de un voicemail asociado a la extensión del usuario autenticado. Con estos datos se puede realizar la descarga de un voicemail siguiendo los pasos descritos en el “Manual de Usuario”.
      Parámetros de entrada:
        - file: (string) nombre del archivo voicemail.

      Parámetros de salida:
        - audio: (string) contenido del archivo de audio, codificado en base64.
        - contentType: (string) tipo de contenido del archivo.
        - size: (string) tamaño del archivo.

Clase para generar el WSDL

Para la generación del WSDL se creó una clase llamada WSDLcreator.class.php

Esta clase consta con las siguientes funciones:

- **generate:** Esta función se encarga de generar el WSDL.
  Parámetros que recibe:
    - arrFP: (array) arreglo con los puntos funcionales del WSDL.

  Retorna un bool, true en caso de haberse generado el WSDL, caso contrario false.

- **enableCache:** Habilita la cache.
  Retorna un bool, true si la cache fue activada y se escribió el archivo exitosamente, caso contrario false.

- **export:** Esta función realiza la exportación del WSDL.
  Parámetros que recibe:
    - mode: (string) los modos pueden ser: 'download', 'file' o 'print', por default este parámetro es 'print'.
    - targetFile: (string) si modo es 'file' esta variable debe ser la ruta del archivo.

- **getError:** Retorna el atributo error de la clase.

- **getWSDL:** Obtiene el WSDL como string y lo retorna.

- **createWSDL_TPL:** Crea la plantilla WSDL como un objeto XML.
  Retorna un bool, true si fue creado exitosamente, caso contrario false.

- **setMessage:** Crea la entidad “message” en el WSDL.
  Parámetros que recibe:
  - function: (string) nombre del punto funcional para colocarlo en la entidad wsdl:message.
  - arrParams: (array) arreglo de parámetros de entrada y salida del punto funcional.

  Retorna un bool, true si la entidad “message” fue creada, caso contrario false.

- **setSchemaTypes:** Crea la entidad “element” en el WSDL.
  Parámetros que recibe:
  - nameElement: (string) nombre del elemento en el esquema.
  - arrParamType: (array) arreglo con los tipos de parámetros de entrada y salida.

  Retorna un bool, true si la entidad “element” fue creada, caso contrario false.

- **addElements_SchemaTypes:**
  Parámetros que recibe:
  - objElement: (object) elemento tipo nodo para crear recursivamente elementos “complexType”.
  - arrParamType: (array) arreglo con los tipos de parámetros de entrada y salida.

  Retorun un bool, true si la entidad fue creada, caso contrario false.

- **isValid_DataType:** Verifica si el tipo de dato es válido.
  Parámetros que recibe:
  - datatype: (string) tipo de dato a validar.
  - param: (string) nombre del parámetro que tiene el tipo de dato a validar.

  Retorna un string con el nombre correcto de tipo de dato o false si el tipo de dato no existe.

- **setPortType_Operation:** Crea la entidad “Operation/PortType” en el WSDL.
  Parámetros que recibe:
  - function: (string) nombre del punto funcional para la entidad wsdl:operation.

  Retorna un bool, true si la entidad “Operation/PortType” fue creada, caso contrario false.

- **setBinding_Operation:** Crea la entidad “Operation/Binding” en el WSDL.
  Parámetros que recibe:
  - function: (string) nombre del punto funcional para la entidad wsdl:operation sección binding.

  Retorna un bool, true si la entidad “Operation/Binding” fue creada, caso contrario false.

Clase para el manejo de SOAP

Para el manejo de SOAP se creó una clase llamada SOAPhandler.class.php

Esta clase consta con las siguientes funciones:

- **exportWSDL:** Esta función exporta el WSDL, si es que todavía no se lo ha hecho.
  Parámetros que recibe:
  - mode: (string) los modos pueden ser: 'download', 'file' o 'print', por default este parámetro es 'print'.
  - targetFile: (string) si modo es 'file' esta variable debe ser la ruta del archivo.

  Retorna un bool, true si la exportación se realizó con éxito, caso contrario false.

- **authentication:** Esta función verifica que el usuario en la variable $_SERVER['PHP_AUTH_USER'] esté correctamente autenticado.
  Retorna un bool, true si la autenticación es correcta, caso contrario false.

- **execute:** Este procedimiento inicia el servidor SOAP.
  Parámetros que recibe:
  - cache: (bool) true para habilitar la cache, caso contrario false.

- **getError:** Esta función retorna el error ocurrido (en caso de haberlo) y lo lanza como un SOAP fault.

- **createSOAPFault:** Crea un SOAP fault.
  Parámetros que recibe:
  - faultcode: (string) código del SOAP fault.
  - faultmsg: (string) breve descripción del SOAP fault.
  - faultdesc: (string) descripción detallada del SOAP fault.
  - classname: (string) nombre de la clase donde se inició el error.

  Retorna el xml con el SOAP fault.

Interacción entre SOAPHandler y WSDLCreator

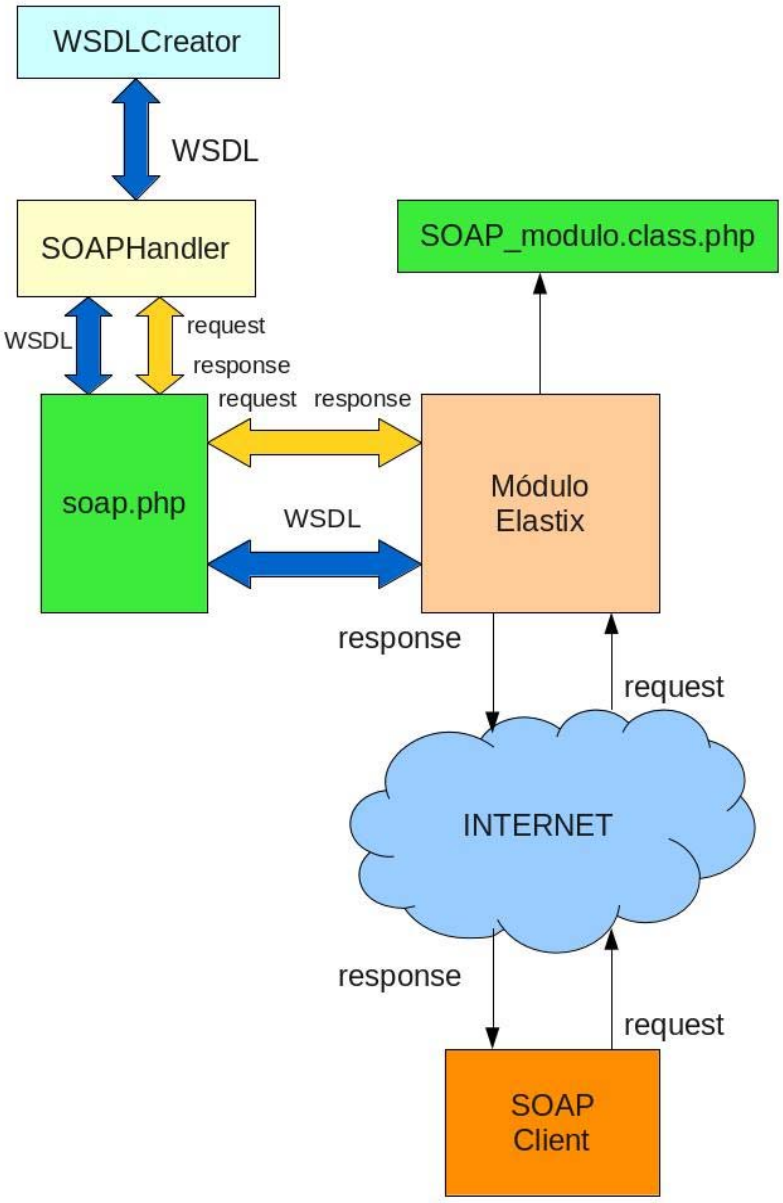
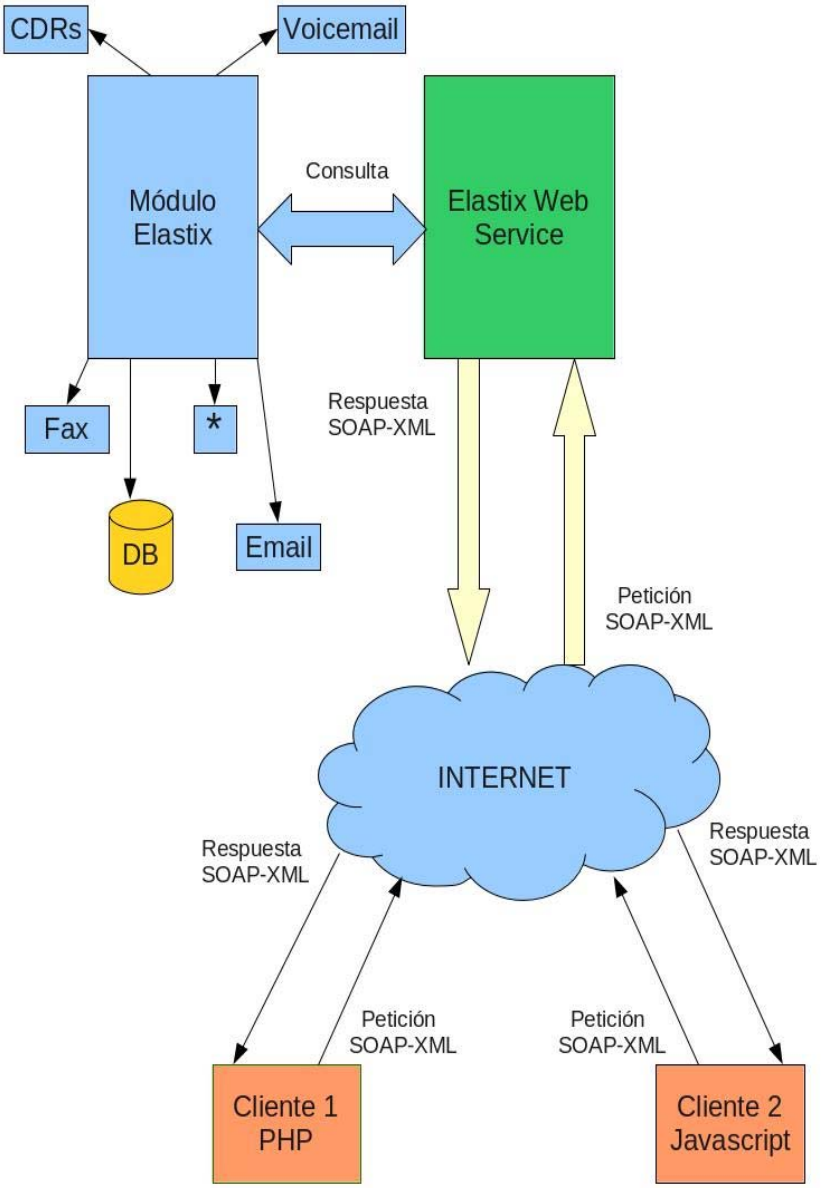


Diagrama de Flujo



No objetivos

Por ahora se mencionan los puntos que no están incluidos en este desarrollo:

- Todos los módulos de Elastix que tengan para hacer exportación se los podría convertir también como Web Services.